

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials (GETI)

Sensor ultrasònic múltiple per a conducció autònoma

MEMÒRIA

30 d'abril de 2019

Autor: Clàudia Cabré Piqueras
Directors: Arnau Doria i Victor Repecho
Convocatòria: Q1 2018/2019



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

El present treball busca solucionar una problemàtica de detecció d'obstacles amb sensors d'ultrasons. El treball s'emmarca en la recerca en control de velocitat adaptatiu (adaptive cruise control en anglès) dut a terme per l'IOC de la UPC.

L'estudi gira entorn la implementació de una solució amb diversos sensors mitjançant un filtre de Kalman, així que fa incís en la formació sobre el funcionament d'aquest.

En una primera instància, s'estudien les bases físiques i matemàtiques del filtre així com els algorismes de processament d'aquest i a continuació es dissenya un filtre per poder solucionar la problemàtica afrontada. Aquest disseny es simula mitjançant el programari Octave per ratificar-ne la fiabilitat.

Finalment es realitza una proposta d'implementació sobre el maquinari i programari amb els quals el sistema treballa actualment emprant el programari Eclipse.

Índex

1	Prefaci	9
1.1	Origen del projecte	9
1.2	Requeriments previs	9
1.3	Estructura del treball	10
2	Introducció	11
2.1	Problemàtica a solucionar	11
2.2	Estat de l'art	11
2.2.1	Triangulació	11
2.2.2	Xarxes neuronals artificials	12
2.2.3	Xarxes bayesianes	13
2.2.4	Filtres de Kalman	14
3	Metodologia emprada: filtre de Kalman	15
3.1	Conceptes bàsics	15
3.2	Assumpcions	16
3.2.1	Linearitat	16
3.2.2	Soroll blanc	16
3.2.3	Distribució normal del soroll	17
3.3	Estructura de la implementació	18
3.3.1	Definició	18
3.3.2	Algorisme de processament	19
4	Disseny del filtre	21
4.1	Consideracions prèvies al disseny	21
4.2	Compliment de les assumpcions de Kalman	23
4.3	Determinació dels coeficients de les matrius	23
4.4	Simulació	24
4.4.1	Simulació d'un sensor amb el disseny proposat	25
4.4.2	Simulació d'un sensor amb el processament mixt	27
4.4.3	Simulació de tres sensors amb el processament mixt	28
5	Implementació	31
5.1	Maquinari emprat	31
5.1.1	Placa STM32F407	31
5.1.2	Sensor HC-SR04	32
5.2	Programari emprat	33
5.2.1	Codi de partida	33
5.2.2	Codi proposat	35
6	Estudi econòmic	37
6.1	Cost de l'estudi	37
6.2	Cost de l'execució	37
6.3	Cost total del projecte	38
7	Estudi sobre l'impacte ambiental	39
7.1	Impacte social	39

7.2 Residus generats	39
8 Conclusions	41
8.1 Contribucions	41
8.2 Desenvolupament i treballs futurs	41
Agraïments	43
Bibliografia	45
Annexos	47
A Codi de la simulació d'un sensor amb el disseny proposat	47
B Codi de la simulació d'un sensor amb el processament mixt	49
C Codi de la simulació de tres sensors amb el processament mixt	51
D Codi de la solució proposada	55

Índex de figures

1	La idea rere la triangulació, Font: [2]	12
2	Estructura d'una xarxa neuronal artificial	13
3	Sistema amb filtre de Kalman, Font: [14]	16
4	Bandes de pas de la densitat de potència espectral, Font: [14]	17
5	Disposició dels sensors	21
6	Resposta en línia recta	21
7	Resposta girant	22
8	Cas 1 per diversos valors de Q	25
9	Cas 2 per diversos valors de Q	26
10	Cas 2 per a $Q=[1e-5]$	27
11	Cas 2 per a un processament mixt	28
12	Circuit simulat	29
13	Ordre d'aparició de l'obstacle	29
14	Tres sensors amb processament mitx, $Q=[1e-5]$	30
15	Tres sensors amb processament mitx, $Q=[1e-4]$	30
16	Esquema de pins de la placa STM32F407 Discovery Kit, Font: diypiandco.com	31
17	Sensor HC-SR04, Font: [18]	32
18	Esquema d'esdeveniments per a la lectura del sensor, Font: [18]	33
19	Esquema d'esdeveniments per a la lectura múltiple dels sensors	35

Índex de taules

1	Dimensions de les variables del sistema discret	18
2	Resum de punts forts i febles	23
3	Cost del maquinari i el programari	37
4	Cost total de l'estudi	37
5	Cost total de l'execució desglossat, Font: IOC	38

1 Prefaci

1.1 Origen del projecte

La manera en que ens movem està canviant. La incorporació de noves fórmules i vehicles en el teixit de la mobilitat fa que es plantegin escenaris fins ara inexistents. Des del disseny íntegre d'una motocicleta elèctrica fins a la gestió de flotes de patinets, l'enginyeria pren un paper actiu i clau en aquest marc de desenvolupament.

Personalment, és una temàtica que sempre m'ha interessat i he format part del projectes de fet a l'ETSEIB e-Ride ETSEIB, que es un equip que construeix una moto elèctrica de competició de manera bianual per a participar a la competició MotoStudent Electric. Alhora les branques d'elèctrica, electrònica i automàtica són les que m'han agradat més al llarg del grau i experiències professionals. En acabar projecte d'e-Ride ETSEIB, tenia ganes de poder ampliar horitzons i no limitar-me a fer un Treball de Final de Grau directament relacionat amb el que havia dissenyat dins l'equip.

Així doncs, per realitzar aquest treball, s'estableix contacte amb l'Institut d'Organització i Control (IOC) de l'Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB).

L'IOC fa recerca, entre d'altres, sobre el control de velocitat adaptatiu. El control de velocitat adaptatiu, o adaptive cruise control (ACC) en anglès, és un sistema de regulació de la velocitat dels vehicles en funció de la distància de seguretat que es vulgui mantenir amb el vehicle del davant.

Aquests estudis es realitzen mitjançant models a petita escala de sistemes amb vehicles proveïts de control de velocitat adaptatiu i una de les problemàtiques que actualment afronta l'IOC és la de la detecció de l'esmentada distància de seguretat. Els prototips que a hores d'ara es troben en funcionament s'ajuden d'un sol sensor d'ultrasons per tal de realitzar la lectura. L'objectiu d'aquest treball és d'implantació d'un sistema de sensors múltiple per tal d'incrementar el rang de detecció.

1.2 Requeriments previs

Per tal d'elaborar aquest treball calen coneixements previs en:

- Familiarització amb plaques de microcontroladors per prototipatge, com ara Arduino o ESP32.
- Nocions bàsiques l'estructura del programari dels microcontroladors: tasques, interrupcions, comptadors, etc.
- Programació a nivell bàsic.
- Simulació amb scripts d'extensió m.

1.3 Estructura del treball

El treball s'estructura en tres grans blocs.

El primer, comprès pel segon capítol, fa un repàs de la problemàtica actual i introdueix un breu anàlisi de metodologies avaluades per tal de solucionar-la.

El segon, que inclou el tercer i quart capítol, agrupa tot el procés de disseny conceptual i simulació de la solució. Al primer capítol del bloc, el tercer, es detalla la base teòrica de la metodologia emprada. A continuació, al quart capítol, es proposa el disseny de la solució sobre aquesta metodologia.

Finalment, el capítol cinquè comprèn un anàlisi del maquinari i programari del sistema per tal de adaptar el filtre dissenyat a la realitat material del problema.

2 Introducció

2.1 Problemàtica a solucionar

El problema que es presenta en començar aquest treball és que el sensor d'ultrasons que hi ha present en els vehicles amb els quals l'IOC treball actualment no efectua una mesura amb prou angle d'obertura com per detectar qualsevol obstacle en qualsevol circumstància.

Així doncs, es proposa fer servei de tres sensors d'ultrasons i fusionar les seves dades per tal de augmentar el rang de mesura. A partir d'aquí, s'estudien diverses metodologies per a fer aquesta mescla de dades i poder arribar a solucionar aquest problema.

2.2 Estat de l'art

Per tal de decidir com tractar les sortides dels sensors es fa una recerca en metodologies emprades per a fusionar dades de sensors. En primer lloc s'opta per fer una busca àmplia en l'àmbit de fusió de dades. No obstant, ràpidament s'observa que cal acotar la cerca a aplicacions concretes. Així doncs, finalment es realitza un estudi de tecnologies emprades en condicions similars, és a dir, sensors d'ultrasons o infrarojos per a aplicacions d'automoció; en aquest apartat se'n plasma l'estat de l'art.

2.2.1 Triangulació

Generalització

La triangulació és un mètode emprat la trigonometria per a poder determinar la posició d'un punt o la distància a aquest. De manera general, la triangulació es basa en el cobriment d'una superfície mitjançant triangles, de manera que si dos triangles intersequen entre si tenen un costat complet o un vèrtex en comú.[1]

Amb el conjunt d'equacions obtingudes dels triangles generats i les relacions entre ells, s'obté un sistema el qual servirà per obtenir la solució desitjada.

Aplicació concreta

Pel que fa a detecció d'objectes amb sensors d'ultrasons, assumint que un mateix cos retornarà una senyal si dos sensors es disparen des de quasi la mateixa posició, és possible estimar la localització del cos mitjançant aquesta tècnica. Donats dos sensors que llegint un mateix cos, l'objectiu es trobarà en la intersecció dels dos arcs de lectura tal i com s'observa a la figura 1 i resolent el sistema d'equacions (1).

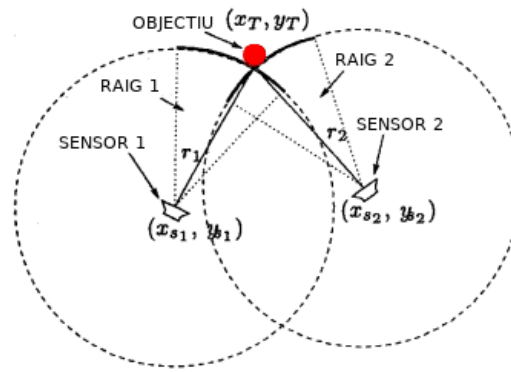


Figura 1: La idea rere la triangulació, Font: [2]

$$\begin{aligned} (x_T - x_{s1})^2 + (y_T - y_{s1})^2 &= r_1^2 \\ (x_T - x_{s2})^2 + (y_T - y_{s2})^2 &= r_2^2 \end{aligned} \quad (1)$$

De la triangulació en destaca que és el mètode més senzill dels valorats, entenent que la relació entre la lectura del sensor (entrada del sistema) i la sortida estimada és la més clara i té un càlcul directe.

No obstant això, la implementació amb tres sensors requeriria més complexitat de càlcul o, per contra, limitacions de processament com per exemple calcular la triangulació només amb els sensors que es troben dos a dos.

Aquest mètode supliria el soroll induït pels sensors d'ultrasons mitjançant un període de mostreig més curt que permetés obtenir una tendència general del senyal.

2.2.2 Xarxes neuronals artificials

Generalització

Les xarxes neuronals artificials (conegudes també com a artificial neural networks o ANN en anglès) s'emmirallen en l'estructura cervell humà [3]. Així doncs, són un tipus d'algorismes formats de diversos nodes amb diverses capes d'operacions ocultes, que rebent certes entrades i relacionant-se entre ells emetran una sortida diferent, tal com si de neurones es tractessin tal i com es representa a la figura 2. D'aquesta manera, les xarxes neuronals artificials pretenen ser una manera eficient de resolució de problemes complexos d'acord amb la intenció de dividir el problema en petites parts per a fer-lo més senzill.

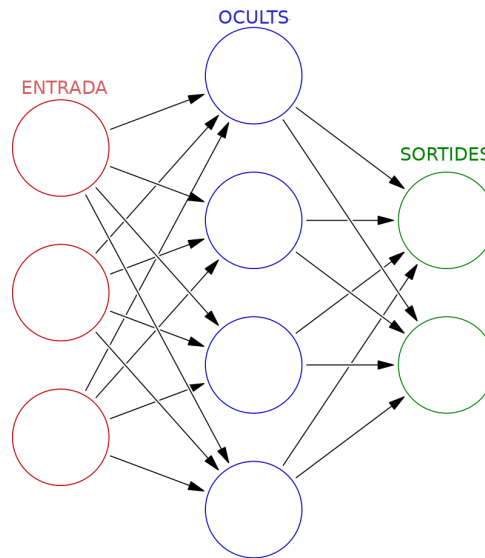


Figura 2: Estructura d'una xarxa neuronal artificial

La relació entre entrades i sortides en una xarxa neuronal artificial vindrà donada per una sèrie de paràmetres que seran modificats en base a un algorisme d'aprenentatge que es proveirà de registres de dades per restablir aquestes constants de càlcul. D'aquest procés, se n'obtindrà una xarxa per a la aplicació concreta amb la que es vulgui treballar. Derivat de aquest procés d'aprenentatge, un dels punts forts de les xarxes neuronals és que les entrades poden presentar correlació entre elles; i, en conseqüència, la xarxa assumir aquesta dependència.

Aplicació concreta

Treballant amb sensors d'ultrasons, les entrades de la xarxa neuronal artificial seran les lectures dels sensors. Com a sortida, se n'obtindrà la distància mitjana a l'obstacle que calgui identificar. Pel que fa a la estructura interna de la capa, la quantitat de capes de processament de la xarxa influeix en el comportament de la sortida de manera.

Referent a la detecció d'obstacles, és un mètode que s'aplica per la necessitat de suplir una mala lectura amb angles oberts; ja que, per exemple, la triangulació pot donar error en aquests casos. [4]

2.2.3 Xarxes bayesianes

Generalització

Les xarxes bayesianes es basen en la interpretació Bayesiana de la probabilitat. Aquesta entén la probabilitat no com una tendència o una freqüència sinó com una previsió raonable derivada de l'estat del coneixement sobre el fenomen d'estudi. Així doncs, les xarxes bayesianes són un model probabilístic gràfic que prediu la probabilitat de que una de moltes causes possibles sigui el factor principal contribuent a la sortida. De la mateixa manera que les xarxes neuronals artificials, es proveeix de algorismes d'aprenentatge per a formar-se. Així doncs, sovint s'aproximen com a xarxes neuronals artificials ja que no tenen processos ocults sinó que són una

representació directa de com es comporta el sistema estadístic tot i que justament per aquest motiu les entrades han de ser necessàriament independents entre elles.[5, 6]

Aplicació concreta

La implementació d'aquest sistema, obviant les diferències de plantejament de base, serà força semblant a la de les xarxes neuronals artificials. Cal definir l'estructura de la xarxa i elaborar un algorisme perquè pugui aprendre en base a una sèrie d'experiments ja realitzats. Un altre cop, la sortida esperada serà un valor de distància que agrupi la lectura dels diversos sensors.[7]

2.2.4 Filtres de Kalman

Generalització

De la mateixa manera que un filtre passa-baixa filtra les freqüències més altes obtenint una senyal filtrada composta sols per freqüències baixes, un filtre de Kalman filtra el soroll o la incertesa per a obtenir l'estat de la senyal desitjat. D'una altra manera, un filtre de Kalman es pot veure com un observador de Luenberger amb soroll a les mesures i entrades. Els filtres de Kalman estan pensats per a operar en sistemes en forma d'espai d'estats lineal, de manera que per treballar-hi les variables i constants s'expressaran en matrius amb les quals s'operarà per a obtenir la sortida desitjada.

Aplicació concreta

De manera concreta, s'usen tant amb diversos senyals d'ultrasons com amb la mescla de diversos sensors. Un exemple d'aplicació molt recurrent que empra els filtres de Kalman junt amb un altre sensor és la mescla d'aquest amb un sensor d'infrarrojos [8, 9]. De tota manera, com s'ha esmentat, també pot processar senyals només d'ultradons amb la qual cosa la sortida obtinguda serà la senyal d'entrada sense aquesta incertesa. [10, 11, 12, 13]

3 Metodologia emprada: filtre de Kalman

Abans de començar el capítol es fa incís en perquè el filtre de Kalman és la metodologia escollida. En primer lloc, d'entrada hi ha una voluntat expressa d'aprendre com funcionen aquest tipus de filtres. Així doncs, un dels motius cabdals és que, donada l'oportunitat presentada, aprofitar per aprendre sobre aquesta metodologia.

En segon lloc, i en detriment a la triangulació, s'observa que el que es vol obtenir com a sortida del processament és un escalar. D'aquesta manera, un sistema com la triangulació pot ser més útil per aplicacions en les que es vulgui realitzar un mapeig de la zona o altri.

En tercer lloc i per últim, es valors que les xarxes bayesianes i les xarxes neuronals artificials són algorismes excessivament complexos per a una aplicació d'aquest caire.

Així doncs, la metodologia de resolució triada és el filtre de Kalman.

Un cop exposats els motius de la tria de la metodologia, comença el capítol. Prèviament, aclarir que l'objectiu d'aquest apartat no és tant la comprensió en profunditat de la base matemàtica del filtre sinó un repàs breu de la teoria i una introducció a la implementació d'aquest.

3.1 Conceptes bàsics

Un filtre de Kalman és, en essència, una algorisme que busca processar les dades de forma òptima respecte uns criteris establerts.

Una vessant del que s'entén per òptim, és la seva capacitat per incorporar tota la informació que se li pugui donar. Pot processar mesures independentment de la seva precisió, coneixement sobre la dinàmica del sistema i la dinàmica dels seus aparells de mesura; així com la definició estadística del soroll del sistema, errors, incerteses i també qualsevol informació sobre les condicions inicials de les variables d'interès. Si la aplicació ho requereix, un filtre de Kalman seria una eina idònia per processar tota aquesta informació.

Alhora, es fa referència al filtre com a recursiu ja que, a diferència d'altres mètodes de processament, un filtre de Kalman no requereix d'un històric de dades per tal de funcionar. Aquesta característica pren una gran importància a l'hora d'implementar-lo.

La figura 3 mostra una situació típica en la qual és avantatjós emprar un filtre de Kalman. El sistema consta d'unes variables de control conegudes i els sistemes de mesura donen el valor de les quantitats mesurades. Analitzant els components del sistema d'equacions en concret: El vector d'estats x és comprès dels valors que son estimats pel filtre. És a dir, el que es vulgui aconseguir filtrant. Sovint, però, pot incloure altres ítems que siguin necessaris per a calcular l'estat del sistema. Per exemple, si es vol determinar la posició d'un punt en un gràfic sabent el valor de la seva segona derivada avaluada en el mateix punt, segurament també calgui afegir al vector d'estats la primera derivada. El sortides son tot el que es coneix explícitament del sistema físic.

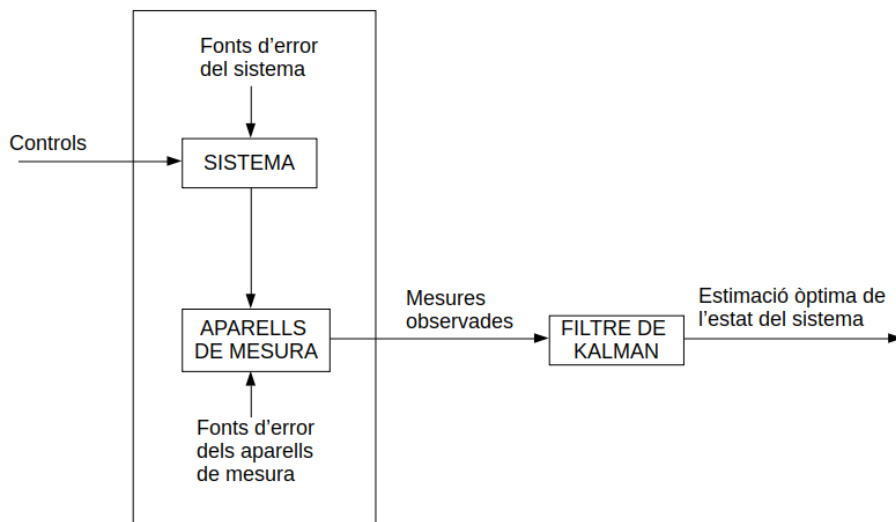


Figura 3: Sistema amb filtre de Kalman, Font: [14]

No obstant això, sovint és necessari el coneixement d'altres variables que descriuen l'estat del sistema que no poden ser mesurades directament i es requereix de mecanismes per generar-les. El filtre de Kalman combina tota la informació que es pot obtenir del sistema per a estimar les variables necessàries de manera que alhora en minimitza l'error estadísticament.

Al paràgraf anterior, es fa referència al que és òptim. Per un filtre de Kalman, aquest òptim significa minimitzar l'error respecte al sistema sense filtre.

3.2 Assumpcions

Hi ha tres assumpcions bàsiques per a formular un filtre de Kalman. A continuació, s'exposen:

3.2.1 Linearitat

Per a treballar amb un filtre de Kalman és necessari que el model del sistema sigui lineal. Els sistemes lineals són més complerts i pràctics que els no-lineals en tant que un sistema no-lineal sovint s'afronta, des de l'enginyeria, linealitzant-lo de manera que s'assumeix una pertorbació o error en el model final. A banda, els sistemes lineals són més fàcils de manipular amb les eines de l'enginyeria.

Si bé es cert que hi ha maneres d'ampliar el filtre de Kalman a models no-lineals, és preferible treballar des de primera instància amb un model que compleixi amb aquest requeriment.

3.2.2 Soroll blanc

Que un soroll sigui blanc implica que el valor del soroll no està correlacionat amb el temps, és a dir, que el seu valor no depèn de l'instant en que es llegeixi. Alhora també implica que aquest

soroll té la mateixa incidència en totes les freqüències.

Físicament, el soroll blanc és una banda de pas de soroll blanc ja que és materialment impossible tenir la mateixa incidència en totes les freqüències. De manera general i vist gràficament, una senyal es veurà afectat per aquesta banda de pas i com es veu a la figura 4, enfrontant l'espectre de freqüències amb la densitat de potència espectral de cada senyal. La banda de pas del soroll blanc té més densitat de potència relativa més enllà de les freqüències en les quals s'emetrà la senyal i romandrà constant dins aquest mateix rang de freqüències.

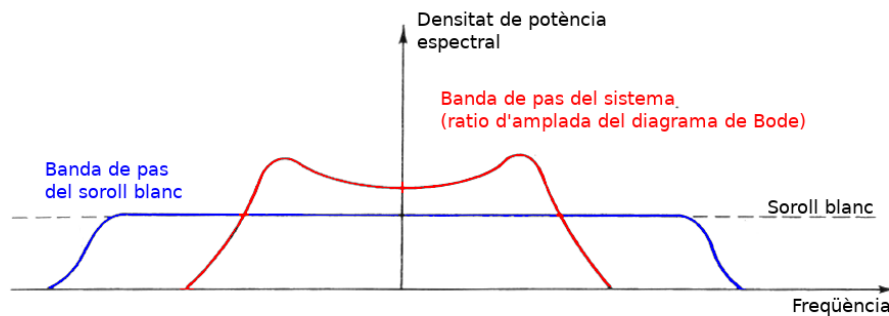


Figura 4: Bandes de pas de la densitat de potència espectral, Font: [14]

Essent constant en el rang de treball de la senyal, es justifica l'ús d'un sistema amb soroll blanc ja que a efectes pràctics compleix els requeriments de constància i no-correlació amb el temps. No obstant això, en cas de no ser-ho es podrà implementar un filtre previ per tal de modular-lo.

3.2.3 Distribució normal del soroll

El soroll present en les senyals tractades per el filtre ha de regir-se per una distribució normal. De manera anàloga al soroll blanc, una distribució normal permetrà que el filtre sigui més fàcil de tractar matemàticament.

Aquesta assumptió pot ser justificada físicament pel fet que el soroll és causat, normalment, per diverses fonts petites. Matemàticament, i pel Teorema del Valor Final, quan un número de variables aleatòries i independents se sumen la suma d'aquestes tendeix a una formar una distribució normal.

Un cop exposades les assumpcions, si bé es cert que poden semblar massa restrictives, cal fer incís en que estan directament alineades amb l'objectiu de que el model es pugui desenvolupar. Si l'objectiu de l'estudi fos construir un model descriptiu, no caldria limitar-se a un sistema lineal amb soroll blanc regit per una distribució normal sinó que caldria buscar la estructura que mes bé s'adaptés a les dades generades pel sistema real.

3.3 Estructura de la implementació

3.3.1 Definició

A continuació s'exposarà l'estructura d'un filtre de Kalman discret, entenent que si el processament es fa des d'un microcontrolador el temps no serà continu.

D'acord amb l'exposat als punts anteriors, el filtre de Kalman està dissenyat per a operar en sistemes mitjançant l'estructura d'espai d'estats. En tant que l'expressat en minúscula són vectors i en majúscula matrius:

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (2)$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \quad (3)$$

Cadascun dels termes correspon al que es detalla a la Taula 1.

Variable	Descripció	Dimensió
\mathbf{x}	Vector d'estat	$n_x \times 1$
\mathbf{y}	Vector de sortides	$n_y \times 1$
\mathbf{u}	Vector d'entrades	$n_u \times 1$
\mathbf{w}	Vector de processament del soroll	$n_x \times 1$
\mathbf{v}	Vector de mesura del soroll	$n_y \times 1$
\mathbf{F}	Matriu del sistema - Estat	$n_x \times n_x$
\mathbf{G}	Matriu del sistema - Entrades	$n_x \times n_u$
\mathbf{H}	Matriu de l'observador	$n_y \times n_x$

Taula 1: Dimensions de les variables del sistema discret

El nombre d'estats, entrades i sortides del sistema són independents entre si, de manera que les matrius \mathbf{G} i \mathbf{H} poden no ser quadrades. La dinàmica de l'estat està definida per la equació (2) mentre que la equació 3 defineix les equacions de les sortides.

Analitzant els components del sistema d'equacions en concret:

El vector d'estats \mathbf{x} és comprès dels valors que són estimats pel filtre. És a dir, el que es vulgui aconseguir filtrant. Sovint, però, pot incloure altres ítems que siguin necessaris per a calcular l'estat del sistema. Per exemple, si es vol determinar la posició d'un punt en un gràfic sabent el valor de la seva segona derivada avaluada en el mateix punt, segurament també calgui afegir al vector d'estats la primera derivada.

El vector de sortides \mathbf{y} no respon al que surt del filtre com a tal sinó al que s'és capaç de mesurar. Cal expressar les mesures en termes dels estats de manera que es puguin comparar amb les mesures reals. De manera general doncs, el vector de sortides conté els valors que tant poden ser determinats matemàticament a partir de l'estat com a partir d'un sistema de mesura independent.

El vector d'entrades \mathbf{u} conté la informació que requereix el filtre per tal de definir la dinàmica del sistema, es pot entendre com una realimentació. Aquests valors poden ser les mesures directes dels sensors, i en aquest cas, la incertesa o error d'aquestes entrades s'haurà de tenir en compte.

Els termes \mathbf{w} i \mathbf{v} corresponen als vectors de processament i mesura del soroll. Aquests vectors s'empren per modelar la incertesa o soroll de les pròpies equacions.

La incertesa en un sistema es pot manifestar de diverses maneres. En primer lloc amb l'error derivat dels aparells de mesura, que n'és el seu ús comú. No obstant, també pot incloure l'error de les pròpies equacions que en defineixen la dinàmica: les equacions derivades d'un principi físic tenen un error negligible mentre que amb altres definicions no es tindrà la mateixa confiança.

Per últim, les matrius \mathbf{F} , \mathbf{G} i \mathbf{H} depenen directament del problema plantejat ja que s'empren per representar les equacions en forma de sistema linial d'estats i entrades.

La matriu \mathbf{F} conté els coeficients de l'estat (2). La matriu \mathbf{H} conté, de manera anàloga a \mathbf{F} , els corresponents a les sortides (3). Finalment, la matriu \mathbf{G} conté els coeficients de les entrades de relimentació. Aquestes matrius, de forma general, poden canviar amb el temps però no poden canviar respecte els estats o les entrades tot i que en molts problemes es defineixen com a constants.

3.3.2 Algorisme de processament

El filtre de Kalman prediu un valor que després és corregit per tal de determinar els estats del filtre. Com s'ha anat esmentant al llarg de la descripció del filtre, la idea principal és fer servir informació sobre la dinàmica del sistema per predir l'estat futur del sistema. És a dir, si es coneix un estat passat i què està fent el sistema es podrà saber l'estat present. A efectes pràctics el concepte del càlcul és prou anàleg al d'un mètode per integració com, per exemple, el mètode d'Euler.

Per tal de resoldre el sistema, però, cal definir quatre matrius més corresponents als valors de covariància entre les diverses variables i als paràmetres del propi filtre, són les matrius \mathbf{P} , \mathbf{Q} , \mathbf{R} i \mathbf{K} . Aquestes matrius no formen part de la definició del filtre com a tal però són necessàries per a la resolució de l'algorisme.

En primer lloc, la matriu \mathbf{P} respon a la covariància amb l'error de l'estat del sistema. En segon lloc, la matriu \mathbf{Q} fa referència a la covariància amb el soroll degut a el processament. La matriu \mathbf{R} es refereix a la covariància amb la mesura del soroll. Per últim, la matriu \mathbf{K} correspon a la matriu de guany del filtre de Kalman.

Per iniciar el procés recursiu, es parteix des d'un estat estimat $\hat{\mathbf{x}}_0$ i una matriu de covariància de l'error de l'estat inicial, \mathbf{P}_0 .

En primer lloc, el vector de l'estat és predit a partir de l'equació de la dinàmica del sistema (2):

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1} \quad (4)$$

on $\hat{\mathbf{x}}_{k|k-1}$ és l'estat predit i $\hat{\mathbf{x}}_{k-1}$ és l'anterior estat predit. La notació « $k|k-1$ » es llegeix com a « k donada $k-1$ » i fa referència a que l'estat definit a l'instant discret k depèn directament de l'instant previ, $k-1$.

Noti's que a diferència de l'equació 2 vista a la descripció del filtre, el vector \mathbf{w} no apareix. Això es deu a que, a efectes pràctics, s'assumeix que \mathbf{w} és un error aleatori amb una mitja de valor zero. Així doncs, la informació de \mathbf{w} s'empra per a determinar informació de la matriu de covariància amb el soroll degut a el processament, \mathbf{Q} . Passa el mateix de manera anàloga amb

el vector \mathbf{v} i la matriu \mathbf{R} .

Seguint amb l'algorisme, cal predir la matriu de covariància de l'error de l'estat inicial, \mathbf{P}_0 , emprant:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \quad (5)$$

on $\hat{\mathbf{P}}_{k|k-1}$ és la matriu predita i $\hat{\mathbf{P}}_{k-1}$ és la anterior matriu predita tal i com passava amb el vector d'estat a l'equació (4). Un cop s'han calculat els valors predits, s'obté la matriu de guany del filtre de Kalman \mathbf{K}_k :

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1} \quad (6)$$

Seguidament, s'actualitza el vector d'estat. Això es farà comparant la mesura real de la sortida \mathbf{z}_k amb la sortida predita $\mathbf{H}_k \hat{\mathbf{x}}_{k-1}$ (que també es pot expressar com a $\hat{\mathbf{y}}_{k|k-1}$) i multiplicant-ho per la matriu de guany del filtre de Kalman per tal de corregir la predicció en la mesura correcta:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \right) \quad (7)$$

De la mateixa manera, s'actualitza a continuació la matriu \mathbf{P} :

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (8)$$

on \mathbf{I} és la matriu identitat. Donats aquests passos, es dona per finalitzada l'actualització de l'estat.

4 Disseny del filtre

En aquest apartat s'explica com s'han adaptat de manera teòrica els filtres de Kalman a aquesta aplicació en concret.

4.1 Consideracions prèvies al disseny

Si bé es cert que els filtres de Kalman poden assimilar molta informació alhora, cal pensar en quina és la manera més òptima per implementar-los per tal que el codi sigui el més ràpid possible.

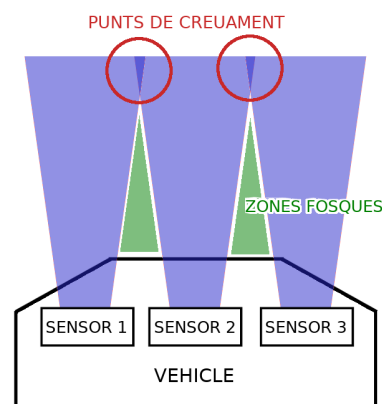


Figura 5: Disposició dels sensors

Així doncs, s'estudia l'estructura del vehicle i com estan disposats els sensors sobre aquest, tal i com s'esquematitza a la figura (5). Arran d'això, s'observa que:

- Quan el vehicle avança en línia recta i té un altre vehicle davant, per simetria, la distància als sensors 1 i 3 és aproximadament la mateixa (6).

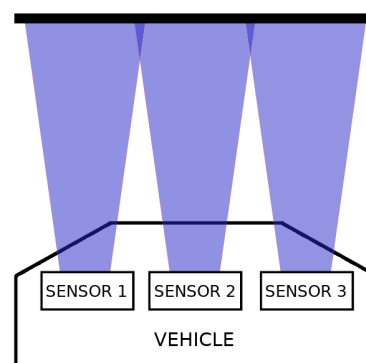


Figura 6: Resposta en línia recta

- Quan el vehicle gira i té un altre vehicle davant hi ha diferències entre les distàncies als sensors 1 i 3 (7).

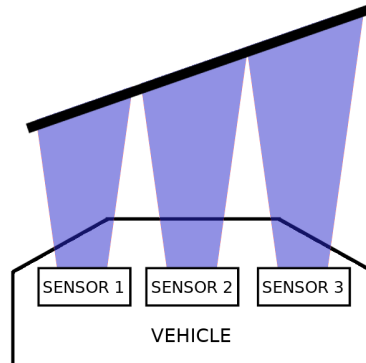


Figura 7: Resposta girant

Vistes aquestes casuístiques, es valoren diverses opcions a l'hora de fer servir el filtre:

- **Opció 1. Filtre únic que engloba totes tres senyals i obté un escalar únic com a resultat:**
Per fer-ho d'aquesta manera cal incloure també en el filtre la relació entre les senyals d'entrada i la sortida. Dit d'una altra manera, aquest tipus d'implementació requereix una operació de, per exemple, triangulació. Si s'implementa d'aquesta manera, l'esquema del processament de les senyals sempre serà el mateix independentment del valor de les senyals rebudes.
- **Opció 2. Filtre segons una lectura prèvia de les distàncies:**
L'objectiu d'una implementació d'aquest tipus és la de processar només les dades que són necessàries. Donades tres senyals de valor semblant, es processa tan sols una senyal ja que s'entén que la resta són anàlogues i no aporten més informació sobre la distàncies. En cas de trobar-se en una corba, es processen les senyals dels dos sensors que ofereixin el valor més baix (s'entén que sensor 2 i ,1 o 3) mitjançant també un càlcul integrat de relació entre entrades i sortides.
- **Opció 3. Filtre de cada senyal individual i processament posterior:**
La raó de valorar aquesta implementació és la de tenir d'entrada un senyal exacte. Després el processament consistiria en comparar totes tres senyals per veure quina és la menor sense risc a causa de error induït pel soroll del sensor. És la opció més senzilla en tant que no té relacions entre senyals dins el propi filtre.

De manera més concisa, els punts forts i febles de cadascuna de les opcions a la taula 2:

Cas	Opció 1	Opció 2	Opció 3
Filtratge del soroll de les senyals	Sí	Sí	Sí
Casuístiques de processament	No	Sí - prèvies	Sí - posteriors

Nombre d'operacions amb el filtre	Processament de tres senyals	Processament d'un o dos senyals	Processament de les tres senyals
Complexitat del càlcul	Relació geomètrica de 3 senyals	Relació geomètrica de màxim 2 senyals i no més en corba	Filtre directe
Resposta davant un obstacle en corba	Cal trobar una relació geomètrica que pugui processar-ho	Es contempla aquest cas	No depèn del filtre
Resposta davant un obstacle que no sigui llegit pels tres sensors	Cal trobar una relació geomètrica que pugui processar-ho	Cal trobar una relació geomètrica que pugui processar-ho	Per assumir-ho de manera individual

Taula 2: Resum de punts forts i febles

S'opta per la 3. El motiu principal és que, de manera general, és l'únic sistema que està més ben preparat davant la detecció d'obstacles que no són rectes ja que pren directament la mesura de cada sensor. Obtenint el valor de cadascuna de les distàncies es podrà caracteritzar qualsevol tipus d'obstacle, mentre que les opcions 1 i 2 serien millors per al seguiment de cossos amb un volum més petit. A banda, amb la opció 3 es garanteixen les assumpcions del filtre de Kalman de manera inequívoca.

Així doncs, i concloent aquest subapartat, es decideix filtrar cadascuna de les senyals individualment entenent que la distància a tenir en compte pel programa sempre serà la més petita de les tres.

4.2 Compliment de les assumpcions de Kalman

Abans de incidir en el disseny com a tal, es fa incís en el compliment de les assumpcions per a poder implementar el filtre de Kalman:

Linearitat

Es compleix a causa que la magnitud de la entrada és la mateixa que la sortida i la intenció del filtre és la de reduir-ne el soroll. No hi ha cap relació geomètrica ni de magnituds físiques entremig que pugui fer que el filtre sigui no lineal.

Soroll blanc i distribució gaussiana

El soroll del sensor HC-SR04 és blanc ja que no depèn del temps. Alhora, aquest presenta una distribució normal en la seva dispersió [17].

4.3 Determinació dels coeficients de les matrius

Per tal de dissenyar el filtre en concret es recorda què es vol filtrar i quines en seran les sortides:

- **A filtrar:** el soroll degut a la dispersió de la distribució normal de la resposta del sensor d'ultrasons, és una incertesa caracteritzable
- **Les sortides:** la lectura del sensor amb la correcció del soroll del filtre

A continuació s'exposaran un per un els termes del filtre que cal . L'ordre d'exposició serà l'ordre d'aparició en l'algorisme de processament (4, 5, 6, 7, 8):

- **Matriu de l'estat del sistema, F:** No hi ha canvi de magnitud entre els estats futurs i passats, els estats representen en tot moment la distància al sensor, així que té un valor d'u.

$$F = [1]$$

- **Matriu de les entrades del sistema, G:** No és un sistema realimentat, així que val zero. De tenir alguna magnitud que corroborés el bon funcionament del sistema, correspondria al guany de la realimentació.

$$G = [0]$$

- **Matriu de covariància amb el soroll degut a el processament, Q:** La matriu Q té un valor arbitrari al principi, ja que s'entén que en desenvolupar un sistema no se sap quin és l'error degut al processament realitzat. No obstant això, cal tenir en compte que el sistema amb el qual es treballa és lineal de per si i a banda el processament de les matrius del filtre de Kalman tan sols consisteix en la eliminació del soroll induït per la falta d'exactitud del sensor. D'això s'extreu que la matriu Q tindrà un valor proper a zero, sinó zero.

$$Q = [0]$$

- **Matriu de l'observador, H:** La magnitud entre el que es mesura i l'estat és la mateixa, és a dir, ambdós valors expressen la distància al sensor. D'aquesta manera, i igual que F, el valor de la matriu H és 1.

$$H = [1]$$

- **Matriu de covariància amb el soroll de la mesura, R:** El sensor té una desviació estàndard de 0,5cm [17], cosa que suposa una variància de 0,025. Com que només hi ha aquest terme amb variància, la matriu de covariància R val 0,025. Aquest valor canviarà depenent de la magnitud amb la qual s'expressi la distància al sensor.

$$R = [0, 025]$$

En aquesta enumeració no s'esmenten la resta de components de l'algorisme, que o bé son els valors predits i les variables internes que canviaran a cada iteració o bé són les dades d'entrada al filtre.

4.4 Simulació

Per tal de corroborar que el plantejament de les matrius és el correcte, es fa un model amb l'ajuda del programari de càlcul Octave. El model es basa en un programa que recorre un fitxer de dades

que segueixen la funció de distribució del sensor d'ultrasons [17] i els aplica seqüencialment la correcció del filtre de Kalman. L'abast de la simulació és emular en primer lloc un sol senyal del sensor per veure si el filtre funciona correctament i a continuació simular un escenari amb els tres que formaran el conjunt finalment.

4.4.1 Simulació d'un sensor amb el disseny proposat

Es programa un fitxer d'Octave que executa els següents passos:

1. Inicialització de variables.
2. Lectura del fitxer i bolcada de dades a una variable.
3. Iteració sobre les dades amb el filtre de Kalman i bolcada de cada iteració sobre una variable.
4. Representació de les senyals a un gràfic.

El fitxer es pot veure a l'Annex A. En primer lloc es treballa sobre un bloc de dades que simula l'aproximació a un obstacle. La distància disminueix de manera uniforme i se suposa velocitat del vehicle constant.

D'entrada, es simula amb una $Q = [1e-11]$ ja que el programari no pot assimilar un valor de 0. Vistos els resultats, es decideix repetir la simulació amb diversos valors de Q , ja que com s'ha esmentat amb anterioritat el valor de Q és, d'entrada, arbitrari. Les sortides del processament es veuen a la figura 8.

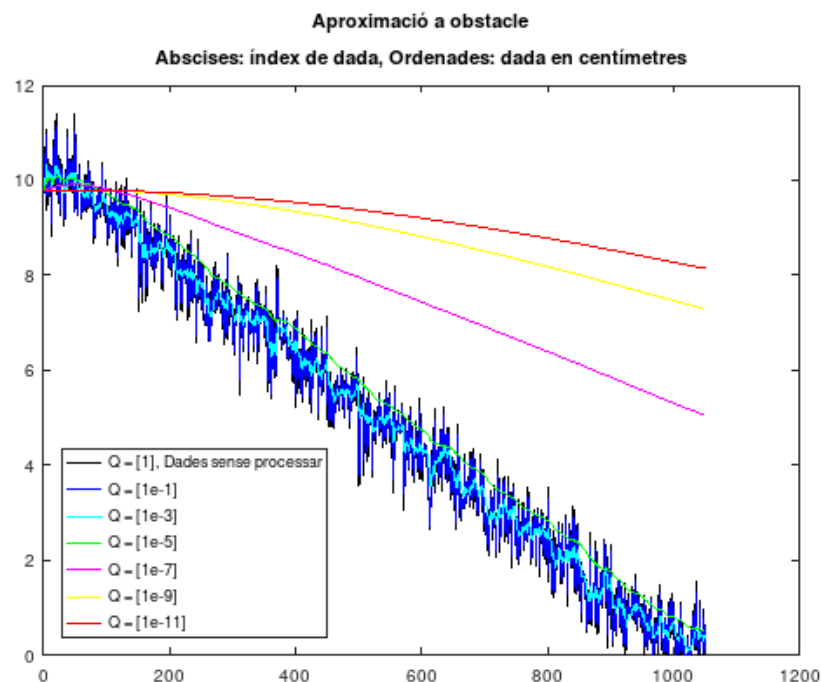


Figura 8: Cas 1 per diversos valors de Q

S'observa clarament com el valor de Q afecta a la dinàmica del filtre. Per valors de Q més alts el filtre és més ràpid però processa menys soroll, de manera contrària passa per valors més baixos.

Aquest comportament té una explicació lligada amb la definició del filtre. La matriu Q és la matriu de covariància amb el soroll degut a el processament, si es fixa el valor de la matriu Q a 1 s'assumeix que l'error que indueix el propi processament és total. D'aquesta manera, els valors que surtin del filtre seran els mateixos que els d'entrada. De manera anàloga passa amb l'altre valor extrem de $Q = [0]$.

No obstant això s'observa que, en aquest cas, per a un cert rang de valors de Q el filtre pot seguir bé la senyal i reduir-ne considerablement la variació.

En segon lloc, es decideix simular com reacciona el sistema davant un obstacle que apareix de forma sobtada. En aquest cas les dades canvien de valor de manera abrupta, considerant que el canvi de lectura és quasi bé instantani suposant el pitjor dels casos. Les sortides del processament es veuen a la figura 9.

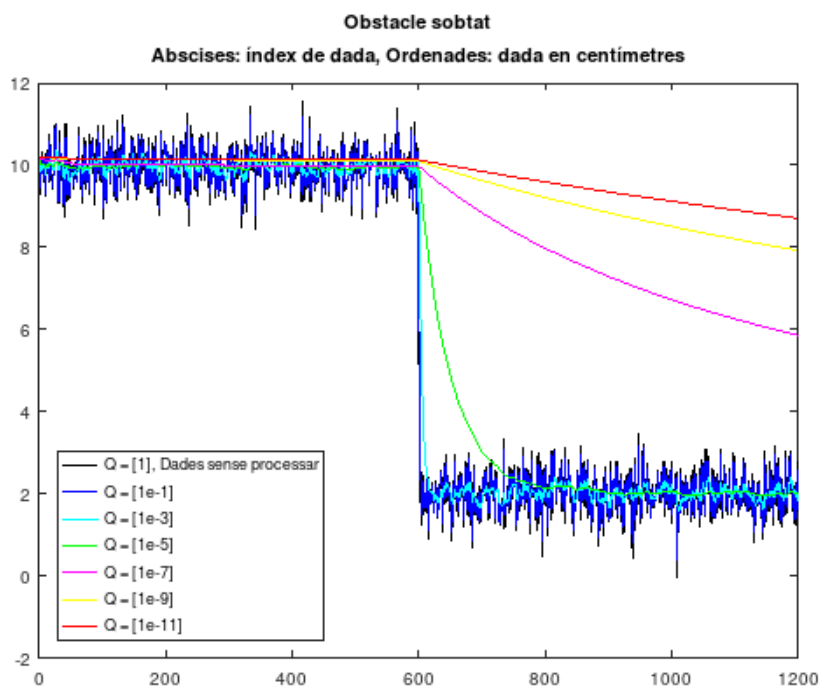
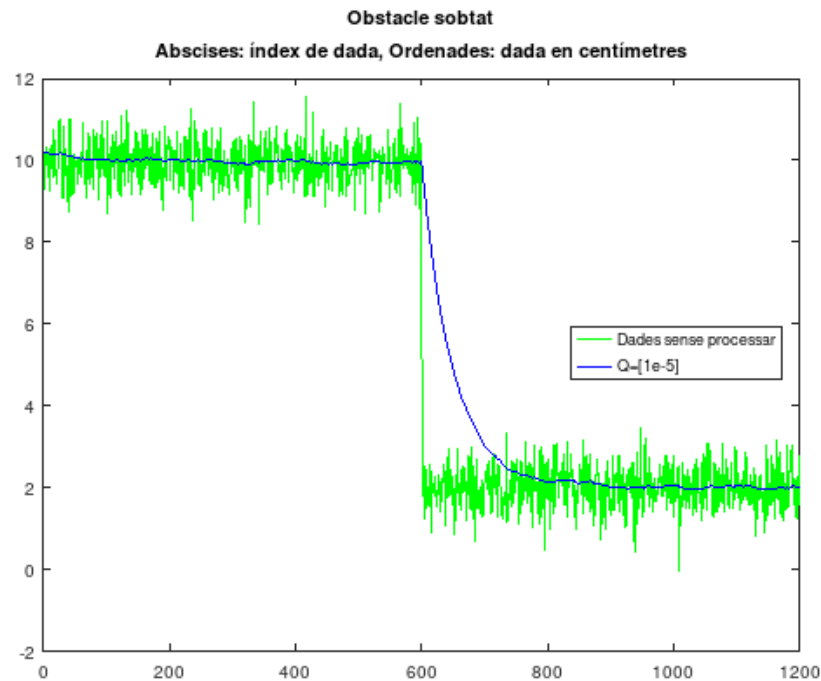


Figura 9: Cas 2 per diversos valors de Q

De manera anàloga al cas de l'obstacle aproximant-se de manera constant, la Q influeix en la dinàmica del filtre.

Seguidament, en concret, es decideix veure el mateix gràfic en concret per a un valor de $Q = [1e-5]$ ja que és un dels valors que en ambdós casos fa que el filtre funcioni millor. La sortida del processament es veu a la figura 10.

Figura 10: Cas 2 per a $Q=[1e-5]$

S'observa que malgrat filtrar la senyal bé en transitori, la resposta davant el canvi és lenta. Això es valora negativament ja que és important que en una aplicació d'aquest caire es pugui reaccionar de manera ràpida davant aquests gradients més grans. Per exemple, extrapolant a un sistema a escala real, és de vital importància que el vehicle pugui frenar a temps en cas que el de davant ho faci. Així doncs, es decideix desenvolupar un processament mixt.

4.4.2 Simulació d'un sensor amb el processament mixt

Es programa un nou fitxer d'Octave que executa els següents passos:

1. Inicialització de variables.
2. Lectura del fitxer i bolcada de dades a una variable.
3. Iteració sobre les dades amb el filtre de Kalman i bolcada de cada iteració sobre una variable: Si la diferència entre l'últim estat estimat i la nova dada llegida és major a un cert escalar, es processa el filtre amb $Q=[1]$. En cas contrari, es processa amb $Q = [1e-5]$.
4. Representació de les senyals a un gràfic.

El fitxer es pot veure a l'Annex B. Com que davant un obstacle que s'aproxima la resposta serà la mateixa que la que s'ha simulat fins ara, es simula només l'aparició d'un obstacle de manera sobtada aquest cop. La simulació es fa amb un valor de $Q=[1e-5]$. La sortida del processament es veu a la figura 11.

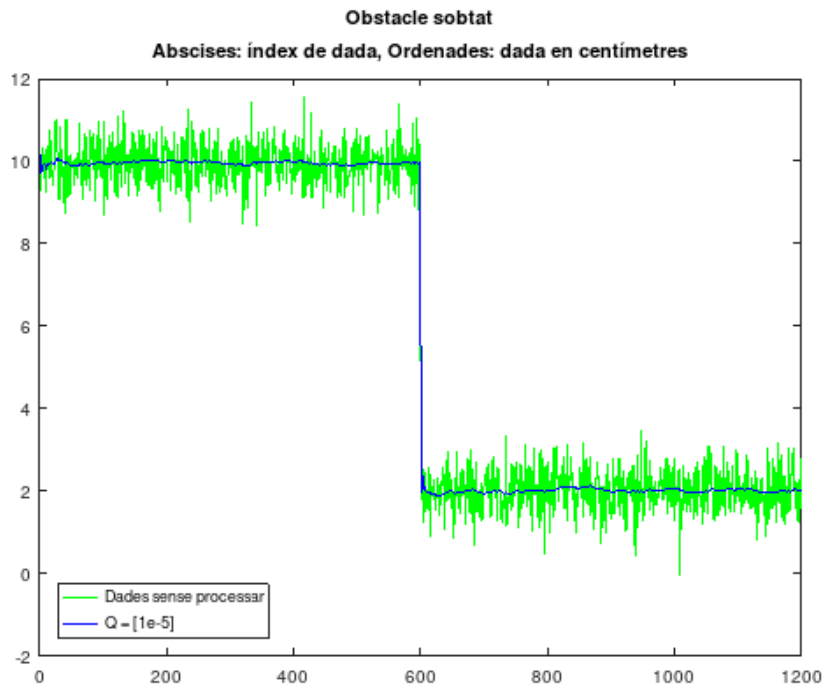


Figura 11: Cas 2 per a un processament mixt

Amb aquest nou processament s'aconsegueix que la resposta sigui ràpida sacrificant la exactitud de la senyal de sortida en l'instant després del gradient, que pateix un lleuger sobrepuig.

Vist la resposta d'aquest model mixt, es decideix que la implementació es farà seguint aquesta casuística. En resum, s'implementa amb les matrius dissenyades i amb la bifurcació de processament entre el coeficient de Q segons la diferència entre l'estat predit i la lectura efectuada.

4.4.3 Simulació de tres sensors amb el processament mixt

De nou, es programa un fitxer d'Octave que executa els següents passos:

1. Inicialització de variables.
2. Lectura de fitxers i bolcada de dades a tres variables, representant les distàncies mesurades en cada cas.
3. Iteració indexada sobre les dades amb el filtre de Kalman i bolcada de cada iteració sobre una variable: S'emmagatzemarà la dada de valor més baix.
4. Representació de les senyals a un gràfic.

El fitxer programat es pot consultar a l'Annex C.

Tractant-se d'una simulació amb els tres sensors, s'han relacionat els tres jocs de dades entre si per tal que es simuli una entrada i sortida de corba amb un vehicle al davant. S'esquematitza el

procés a la figura 12.

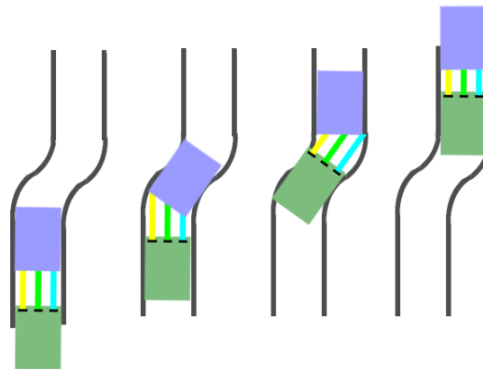


Figura 12: Circuit simulat

Enumerant els elements de la figura; en primer lloc el vehicle que rep les senyals és el verd mentre que el detectat coma a obstacle és el blau. Es numeren els sensors, per tal de referir-s'hi, tal que el número 1 correspon al feix groc, el 2 al verd i el 3 al blau cel.

En un primer moment, abans d'entrar a la corba, els tres sensors llegeixen el mateix valor. Quan el vehicle blau entra a la corba, els sensors 1 i 2 augmenten la distància llegida en relació al 3, que realitza una lectura més propera. Després, quan el blau surt de la corba, les lectures es troben en una situació simètrica a l'anterior: per 2 i 3 la lectura puja mentre que per 1 baixa. Finalment, es torna a circular en línia recta.

A banda d'aquesta dinàmica entre vehicles també es simula s'aparició d'objectes de manera sobtada davant dels tres sensors. L'esquema de les dades es veu a la figura 13.

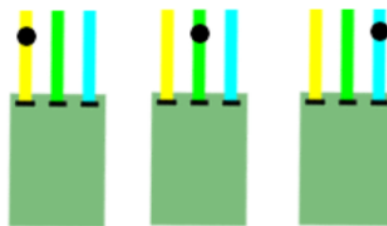


Figura 13: Ordre d'aparició de l'obstacle

Així doncs, es simula el model amb les dades esmentades per a un valor de $Q=[1e-5]$, que és el valor que s'ha considerat més òptim de la simulació amb un sensor. Els resultats de la simulació es veuen a la figura 14.

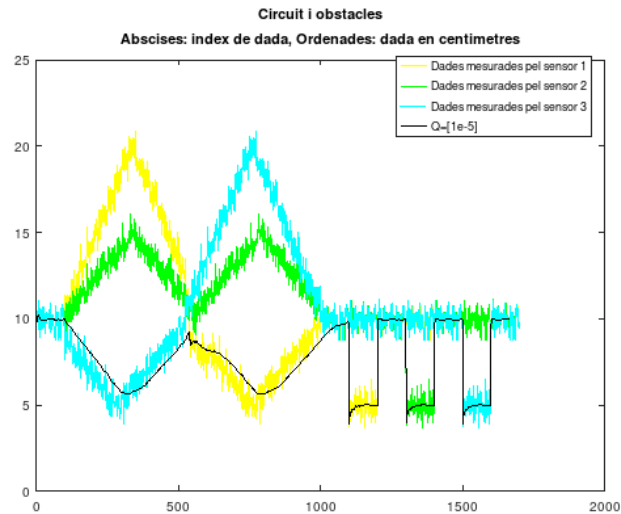


Figura 14: Tres sensors amb processament mitx, $Q=[1e-5]$

En primer lloc, s'observa que el sistema dissenyat és capaç d'adquirir en tot moment la dada filtrada de valor més baix. No obstant això, cal fer notar que, a l'inici de la simulació, quan els canvis són linials, una $Q=[1e-5]$ fa que la distància processada vagi lleugerament més lenta que la real. Per altra banda, els valors s'assimilen bé quan es tracta d'objectes que apareixen de manera sobtada.

Com que la simulació per la Q proposada és lleugerament lenta, es repeteix la simulació per una $Q=[1e-4]$, amb un coeficient de valor més gran. Els resultats de la simulació queden palesos a la figura 15.

En aquest cas, es pot observar que la distància processada va pràcticament a la mateixa velocitat que la real, cosa que es valora positivament malgrat filtrar una menys soroll.

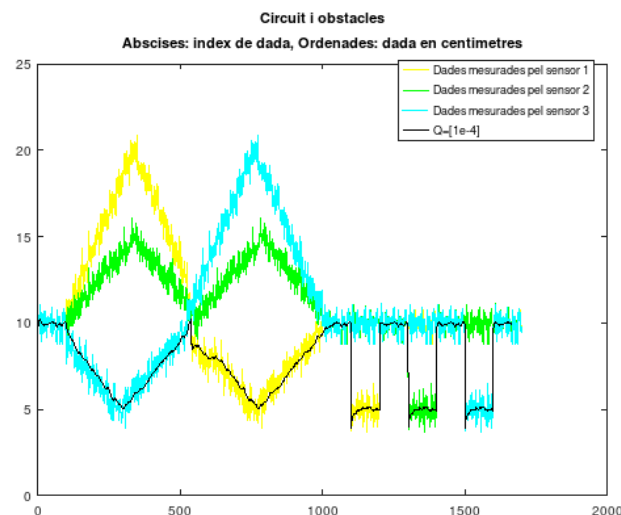


Figura 15: Tres sensors amb processament mitx, $Q=[1e-4]$

5 Implementació

Per tal de traslladar el filtre dissenyat al problema real, cal saber quin es l'entorn en el qual es desenvolupa. Per això es fa un estudi tant del maquinari com del programari que empra el sistema.

5.1 Maquinari emprat

Malgrat el vehicle consta de diversos elements mecànics i perifèrics, aquest punt se centrarà en fer un breu estudi de les característiques de la placa del microcontrolador amb la que es treballa així com del funcionament sensor d'ultrasons emprat.

5.1.1 Placa STM32F407

La placa STM32F407 Discovery Kit és una placa dissenyada per al microcontrolador STM32F407. Destaca per ser un microcontrolador potent, i versàtil, amb capacitat de processar una gran quantitat de perifèrics i tasques. De fet, molts ja formen part de la pròpia placa, com ara un acceleròmetre digital, un micròfon, un altaveu així com diversos leds i botons.

A nivell d'entrades i sortides, compta amb una gran quantitat de pins d'entrada i sortida preparats per diverses aplicacions, tal i com es veu a la imatge 16.

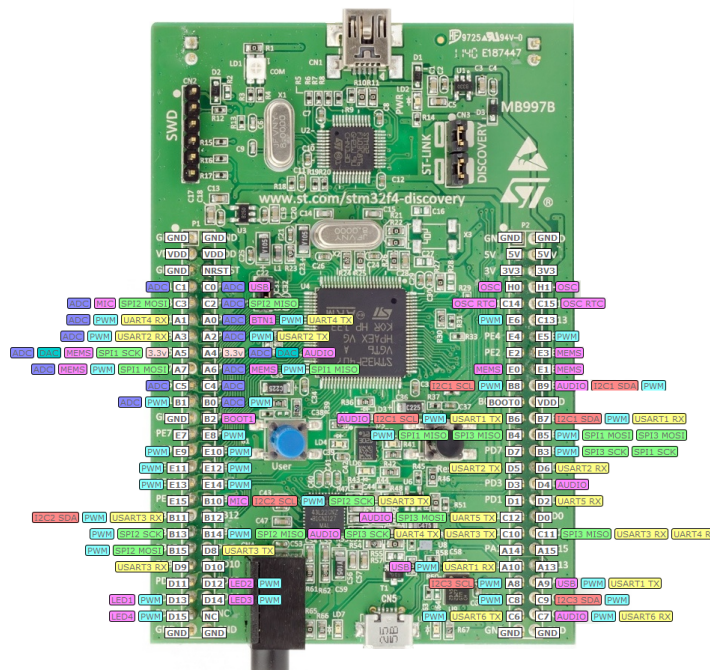


Figura 16: Esquema de pins de la placa STM32F407 Discovery Kit, Font: diypiandco.com

Fent incís en la potència del microcontrolador, es remarca que compta amb diverses infraestruc-

tures per a protocols de comunicació triplicades, com es el cas dels ports d'SPI i I2C. Alhora, pel que fa la versatilitat, la majoria de pins poden proporcionar una sortida PWM (modulació per amplada de pols), cosa que implica que tenen un conversor digital-analògic, o bé poden admetre entrades analògiques, que son aquelles acompanyades pel rètol ADC a la imatge 16.

En resum, és una placa que destaca per poder admetre gran varietat de perifèrics i tenir moltes entrades i sortides que li permeten poder comunicar-s'hi i poder enviar i rebre tant senyals analògiques com digitals.

5.1.2 Sensor HC-SR04

El sensor HC-SR04 és un sensor que funciona en base a ultrasons que pot mesurar des de 2cm fins a 400cm amb una obertura de 15° i amb una exactitud, segons la fulla de dades, de fins a 3mm. [18]

El sensor compta amb 4 pins per connectar-lo a la placa. Aquests són: l'alimentació a 5V, el terra comú a tot els sistema, un pin corresponent al pols de trigger o disparador del pols, i un pin corresponent a l'eco llegit. Pel que fa als dos cilindres que es veuen al sensor, un és l'encarregat d'emetre el pols mentre que l'altre ho és de rebre'l. Tot l'esmentat en aquest paràgraf es pot veure a la figura 17.

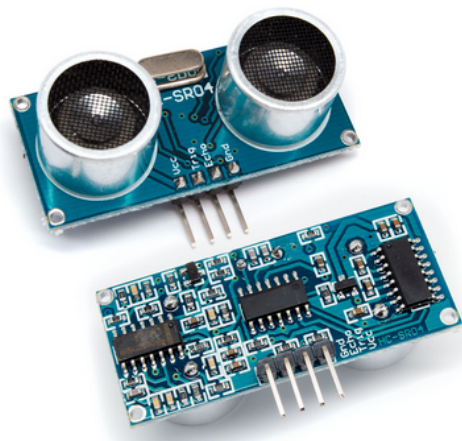


Figura 17: Sensor HC-SR04, Font: [18]

Per tal de realitzar la lectura, els esdeveniments que transcorren son els següents (esquemàticament, a la figura 18):

1. Enviament des del microcontrolador d'una senyal digital amb valor alt durant un mínim de 10us a través del pin trigger.
2. Emissió d'una seqüència de 8 polsos d'ultrasons a 40kHz des del mòdul físic.
3. Recepció de l'eco des del mòdul físic receptor i enviament d'una senyal digital de valor alt al microcontrolador mentre es rebí aquest pols de tornada.

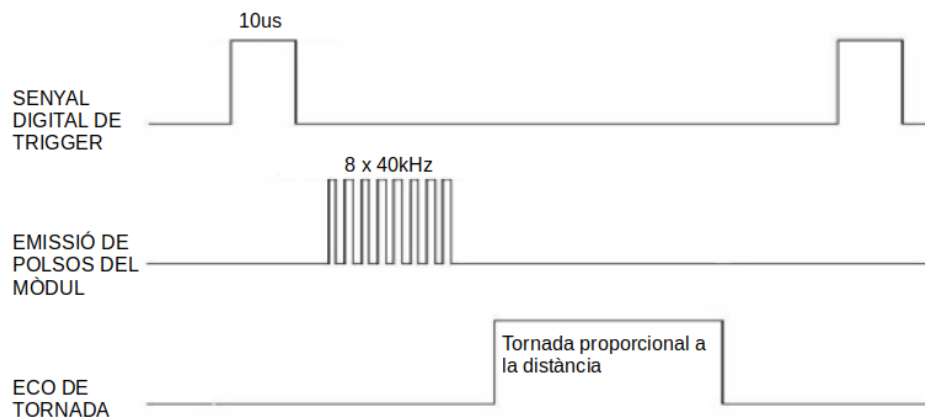


Figura 18: Esquema d'esdeveniments per a la lectura del sensor, Font: [18]

Un cop rebut el senyal d'entrada del receptor de l'eco, la distància es pot calcular de diverses maneres segons si es vol obtenir en centímetres o en polzades.

5.2 Programari emprat

Per tal de poder programar el microcontrolador, es fa ús del programari Eclipse, que és un compilador de C i C++. Sobre aquest programari, s'hi instal·la un entorn de treball (workbench) del propi microcontrolador proporcionat pel fabricant d'aquest.

La interfície del programa permet la visualització de diversos paràmetres alhora com ara la distribució en arbre dels fitxers del programa, la quantitat d'errors i avisos que té el codi o un motor de busca dins el propi espai de treball.

El fet que hi hagi un entorn de treball dissenyat pels propis fabricants fa que sigui molt més ràpid poder programar el microcontrolador, ja que es proveeix d'uns seguit de llibreries que programen el microcontrolador a un nivell més baix per tal que l'usuari pugui treballar amb llenguatges de més alt nivell, com és el cas del C++.

5.2.1 Codi de partida

A l'inici de la implementació, es parteix del codi que ja està implementat sobre els vehicles actuals en funcionament. En aquest punt s'exposa com està configurat el processament relatiu als sensors d'ultrasons.

A l'inici de la implementació, es parteix del codi que ja està implementat sobre els vehicles actuals en funcionament. En aquest punt s'exposa com està configurat el processament relatiu als sensors d'ultrasons. Cal remarcar, però que no es fa incís en enllaç de control que introdueix la variable en aquest sistema, simplement s'estudia la obtenció d'aquest valor escalar.

La característica principal del codi amb el qual funciona actualment el sistema, és que treballa amb comptadors. En concret, el comptador emprat per a la tasca de lectura del sensors d'ultra-

sons és el TIM2, que és un comptador de 16bits amb possibilitat de preescalament. Aquest comptador pot ser emprat per a una gran varietat de propòsits, incloent la mesura de llargada de polsos de senyals (captura d'entrades) o generació d'ones de sortida. Destaca del comptador, que pot assimilar fins a quatre canals independents per a captura d'entrades, captura de sortides o PWM, entre d'altres.

Dins la configuració del comptador, s'ajuda d'un registre comptador i de comparació (TIM2 CCR1) que, quan l'objectiu és el de capturar una entrada, és l'encarregat de guardar el valor del comptador després que hagi passat un tipus d'esdeveniment que prèviament s'haurà programat per detectar. Quan es realitza una captura, la variable bandera CC1IF pren un valor alt i es pot enviar una interrupció al programa, que desarà el valor. [19, 20, 21]

Així doncs, el codi de partida per determinar la distància llegida pel sensor és el següent:

```

1 void TIM2_IRQHandler(void)
2 {
3
4     HAL_GPIO_WritePin(GPIOB,GPIO_PIN_13,GPIO_PIN_SET);
5     if( (TIM2->SR & TIM_SR_CC1IF_Msk) != RESET )
6         {USValue = TIM2->CCR1;}
7
8     IRQ_TIM2_counter++;
9
10    if (IRQ_TIM2_counter==3)
11    {
12
13        IRQ_TIM2_counter=0;
14        USperiod=USValue*count2us;
15        if (USperiod>NULL_USD_MEAS) USd=USperiod*0.02;
16        if (USperiod>MAX_MEAS) USd=(float)MAX_MEAS*0.02;
17        USd_k[1]= 0.4*USd + USd_k[0]*0.6;
18        USd_k[0]=USd_k[1];
19
20        TIM2->EGR = TIM_EGR_UG; // Reinitialize the counter
21    }
22    HAL_GPIO_WritePin(GPIOB,GPIO_PIN_13,GPIO_PIN_RESET);
23 }
```

En primer lloc, cal aclarir que la numeració del codi està fora de context però, no obstant, es manté ja que és útil alhora d'explicar el codi línia per línia:

- (1) Es declara la funció que és l'encarregada de gestionar les interrupcions del TIM2. Que rebi com arguments «void» significa que no necessita cap paràmetre d'entrada per a funcionar.
- (4) S'envia la senyal de trigger al sensor.
- (5) Es comprova si s'ha rebut la interrupció lligada a la rebuda de pols.
- (6) En cas afirmatiu, es desa el valor del temps transcorregut al paràmetre USValue.
- (8 i 10) S'espera fins a rebre tres interrupcions.

- (11) S'actualitza el comptador d'interrupcions.
- (12) Es realitza una primera conversió d'unitats.
- (15) Si el valor mesurat és més gran que el mínim que es vol mesurar (5cm), es pren el valor mesurat.
- (16) Si el valor mesurat és més gran que el màxim que es vol mesurar (200cm), es pren el valor màxim.
- (17) El valor que es té en compte és una suma de productes entre l'últim valor considerat i el valor mesurat.
- (18 – final) Re-inicialització i actualització de paràmetres i posada a un nivell baix del senyal de trigger.

5.2.2 Codi proposat

Partint del codi exposat i el filtre dissenyat es proposa un nou codi per introduir el filtre. A continuació s'exposa l'esquema de funcionalment per blocs (19). El codi final es detalla a l'Annex D.



Figura 19: Esquema d'esdeveniments per a la lectura múltiple dels sensors

En aquest cas, la decisió de quin valor prendrà el coeficient de la matriu Q es farà sempre en referència al sensor amb el qual s'estigui treballant. És a dir, cada sensor tindrà la seva pròpia dinàmica i variarà el coeficient de Q de manera independent.

6 Estudi econòmic

En aquest capítol es realitza una aproximació econòmica del cost de la realització del projecte presentat. El pressupost es distribueix en dos grans blocs: el cost de l'estudi realitzat i el cost d'execució del projecte en si mateix.

6.1 Cost de l'estudi

El cost de l'estudi inclou el cost del maquinari utilitzat, el del programari i també el de la mà d'obra necessària.

En primer lloc, la taula 3 recull les partides del maquinari i del programari emprat en l'elaboració de l'estudi. Les dades d'aquesta taula han sigut extretes de proveïdors d'equips informàtics i de les pàgines web dels fabricants:

Concepte	PVP [euro]	Unitats	Cost total [euro]
Ordinador portàtil	230,36	1	230,36
Perifèrics - ratolí	9,84	1	9,84
Llicències	0	-	0
TOTAL			240,20

Taula 3: Cost del maquinari i el programari

Cal destacar que per a la elaboració d'aquest treball s'ha emprat íntegrament programari lliure, d'aquesta manera el cost de llicències és zero.

D'altra banda la taula cal calcular també recull el cost de la mà d'obra utilitzada. Per tal d'estimar aquest cost, s'ha considerat que un enginyer que just ha acabat el Grau en Enginyeria en Tecnologies Industrials cobra aproximadament uns 10 euros a l'hora. El temps dedicat a l'elaboració del treball és de 300h, contant 25h per crèdit ECTS per 12 crèdits ECTS que corresponen al treball de final de grau. Així doncs, el cost total de la mà d'obra és de **3.000** euros.

Per últim, la taula 4 mostra el cost total de l'estudi realitzat, contemplant els factors esmentats als paràgrafs anteriors:

Concepte	Cost [euro]
Maquinari	240,20
Mà d'obra	3.000,00
TOTAL	3.240,20

Taula 4: Cost total de l'estudi

6.2 Cost de l'execució

El cost de l'execució del projecte inclou tots aquells costos relacionats amb la implementació física del sistema dissenyat. Es detalla a la taula 5.

Concepte	PVP [euros]	Unitats	Cost total [euro]
<i>POTÈNCIA</i>			
DC/DC 12V/5V	6,30	2	12,60
DC/DC 12V/3V3	2,87	2	5,74
Driver	7,00	2	14,01
Motor	28,98	4	115,92
Bateria	32,50	2	65,00
Carregador	69,00	1	69,00
Connectors XT-60	1,10	2	2,20
<i>SENYAL</i>			
Sensor de distància	16,47	6	98,81
Sensor de línia	8,46	2	16,92
Mòdul Wi-Fi	6,53	2	16,07
<i>PLAQUES</i>			
STM32F407	21,99	2	43,97
PCB	10,83	2	21,67
<i>PARTS MECÀNIQUES</i>			
Junta d'L	6,30	2	12,61
Roda	2,14	4	8,57
Adaptador de roda	5,93	4	23,72
Roda boja	6,57	2	13,14
Xassís	20,00	2	40,00
<i>ASSEMBLATGE</i>			
Hores tècnic de laboratori	35,00	6	210,00
TOTAL			786,94

Taula 5: Cost total de l'execució desglossat, Font: IOC

En aquesta taula hi queda palès el cost per a la producció de 2 vehicles. S'entén que per poder testejar el sistema fan falta un mínim de dos vehicles, així que són els contemplats.

6.3 Cost total del projecte

Així doncs, el cost total del projecte és de **4.027,14** euros.

7 Estudi sobre l'impacte ambiental

En aquest punt, s'exposarà l'impacte sobre l'entorn que té aquest treball i el desenvolupament d'aquest. No s'ha volgut limitar aquest apartat a fer un estudi sobre els residus generats per l'aplicació en concret sinó que s'ha destinat també a fer incís en l'impacte que pot tenir socialment.

7.1 Impacte social

Malgrat ser un projecte desenvolupat a petita escala, tota recerca relativa a la mobilitat interurbana i intraurbana i, en conseqüència, al benestar de les persones que conformen una societat, té un impacte positiu en l'entorn.

El control de velocitat adaptatiu s'engloba en aquest marc i té el propòsit de facilitar la convivència entre vehicles. Donada la rapidesa dels avenços tecnològics en el context de la mobilitat, és bo que s'hi faci recerca per tal de trobar-hi noves i millors fórmules que les actuals.

A banda, és positiu que aquests canvis de paradigma neixin des de la pròpia enginyeria. És important que els enginyers siguin sobirans en qüestions que tot i ser directament tecnològiques repercuteixen de manera directa en la societat. Només així, des de la societat i cap a la societat es podrà avançar cap a escenaris millors al present.

Per últim, valorar també l'impacte social de l'ús de programari lliure en aquest treball.

Durant els últims anys, el programari emprat a les escoles d'enginyeria ha tendit cada cop més a ser privatiu. Això ha facilitat dinàmiques de dependència sobre aquestes marques en concret i ha fet que molts cops hagin sobrepassat l'ensenyança programari lliure, que existeix, funciona i és plenament accessible. Aquest treball en concret s'ha realitzat amb una màquina que funciona amb Linux, ha emprat els programari Octave, Eclipse, Gimp i Libre Office, i s'ha maquetat amb el programari LaTeX.

Tot i que no s'espera un gran impacte lligat a aquest treball en concret, sí que hi ha una voluntat de crear precedent en aquest aspecte i aportar recursos a les persones que en un futur vulguin realitzar un treball tenint en compte aquesta premissa.

7.2 Residus generats

Pel que fa als residus generats, cal tenir en compte que tant els prototips com les pistes són elements que es troben en millora continua i per tant no existeix una dinàmica de llençar aquests elements en concret. No obstant, cal tenir en compte els residus que tard o d'hora podran generar.

Pel que fa a les pistes, es valora positivament que estiguin elaborades amb fusta ja que és un material altament reciclable. A banda, abans de reciclar-les, per aquesta aplicació en concret es poden reutilitzar molts cops ja que s'empren per construir les pistes senceres de forma modular.

Referent al vehicle cal diferenciar, a grans trets, entre els elements mecànics i elements electrònics.

En primer lloc, els elements mecànics es comprenen per xassís, rodes, reductors, etc. En general, són un tipus de material que en un taller es poden reutilitzar força vegades abans de llençar-los. A banda, el xassís és d'alumini, que és un material altament reciclable.

Pel que fa als elements electrònics, si bé és cert que molts components són reutilitzables cal tenir en compte que tenen un procés de reciclatge nociu entenent que els encapsulats dificulten la separació dels materials que el conformen. A banda, les PCBs (tant la de la placa STM32407F com la del propi vehicle) no són reciclables un cop se n'ha acabat la vida útil.

Per resumir, no és un projecte que individualment generi una gran quantitat de residus tot i que si que cal tenir en compte que emprà tecnologies que en última instància en generen de nocius. Tot i això també cal tenir en compte que actualment les alternatives reciclables d'aquests components més difícils de reciclar són quasi bé nul·les i que per tant tampoc hi ha una gran capacitat d'actuació en aquest sentit.

8 Conclusions

8.1 Contribucions

En el present Treball de Final de Grau s'ha proposat una metodologia per a fusionar les dades de tres sensors d'ultrasons.

En primer lloc, s'ha estudiat el sistema sobre el qual s'havia de treballar seguit de l'anàlisi de les alternatives que podrien solucionar la problemàtica abordada. Els sistemes valorats han estat la triangulació, les xarxes neuronals artificials i els filtres de Kalman. Finalment s'ha escollit emprar el filtre de Kalman.

En segon lloc, s'ha realitzat un estudi en profunditat de la metodologia escollida. Partint de conceptes bàsics per entendre'n el perquè físic i matemàtic, passant per les condicions que ha de complir el sistema per poder-li-ho implementar i concloent en una exposició de l'estructura del filtre i de l'algorisme de resolució d'aquest.

A continuació s'ha prosseguit a dissenyar un filtre de Kalman per a la aplicació concreta. S'ha partit de les bases que ja s'havien exposat i de les propietats físiques del sensor per tal de definir-ne els paràmetres principals. Per tal de comprovar-ne el funcionament se n'ha programat una simulació la qual ha servit per a millorar el filtre proposat en primera instància.

Finalment, s'ha prosseguit al disseny de la implementació. Per tal d'elaborar-la s'ha realitzat un anàlisi del programari i el maquinari emprats. Un cop estudiats, s'ha elaborat un nou codi que inclou el disseny del filtre de Kalman.

8.2 Desenvolupament i treballs futurs

Pel que fa al desenvolupament del projecte, es valora que s'hauria s'haver repartit el temps d'una manera diferent ja que per a un treball d'aquest tipus hagués estat millor incidir més en la implementació i menys en el disseny. Si bé és cert que en tot moment hi ha hagut una voluntat expressa de aprendre sobre els filtres de Kalman i en cap cas es valora negativament el seu ús, cal dir que són un sistema complex d'entendre i que van restar molt volum de feina a la resta del projecte.

Respecte als treballs futurs, i de mà amb el valorat respecte el desenvolupament, es conclou que un disseny més encarat a la implementació directa amb proves sobre el laboratori seria un enfoc més adequat a la resolució d'aquest problema, des d'una òptica de practicitat i rapidesa. Es valora, doncs, en un futur afrontar aquesta qüestió de manera més mecànica.

No obstant això, l'exactitud oferta pel sistema dissenyat dona peu a noves aplicacions relacionades amb la caracterització d'obstacles, que també és una via a explotar.

Agraïments

En primer lloc, agrair als meus tutors, l'Arnau Dòria i en Víctor Repecho, i a l'IOC per permetre'm desenvolupar un treball de final de grau sobre la seva recerca.

En segon lloc, als meus pares, el David i la Xelo. Per donar-me les eines i ensenyar-me a valorar-les. No només m'heu proporcionat una educació sinó que així m'heu permès poder ser.

En tercer lloc, als companys: a l'e-Ride ETSEIB per tornar-me la fe en l'enginyeria i a l'Assemblea d'estudiants per ensenyar-me dubtar.

I, per últim, a les meves companyes de pis, la Xènia, la Dèlia i la Cristina, per ser-hi.

No hauria arribat a aquí de no ser per vosaltres. Gràcies de tot cor.

Bibliografia

- [1] CHRISTIAN VALQUI, *Triangulaciones y homología simlicial*, Pro Mathematica Vol. XVIII, Nos. 35-36, 2004
- [2] WIJK, P. JENSFELT I H. CHRISTENSEN, *Triangulation based Fusion of Ultrasonic Sensor Data*, Proceedings of the 1998 IEEE International Conference on Robotics Automation Leuven, Belgium * May 1998
- [3] CARLOS GERSHENSON, *Artificial Neural Networks for Beginners*
- [4] KAI-TAI SONG I CHARLES C. CHANG, *Ultrasonic sensor data fusion for enviroment recognition*, proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems Yokohama, Japan July 2630,1993
- [5] P. JARVIS, *Introduction to Bayesian Networks*, Recurs en línia - towardsdatascience.com/
Consultat 14/11/2018
- [6] DIVERSOS AUTORS, *Bayesian probability*, Recurs en línia - en.wikipedia.org/
Consultat 14/11/2018
- [7] ENAS ODAT, JEFF S. SHAMMA, FELLOW I CHRISTIAN CLAUDEL, *Vehicle Classification and Speed Estimation Using Combined Passive Infrared/Ultrasonic Sensors*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS
- [8] LOUIS DROLET, FRANÇOIS MICHAUD I JEAN CÔTÉ, *Adaptable Sensor Fusion Using Multiple Kalman Filters*, Department of Elec. Comp. Engineering Université de Sherbrooke
- [9] KAZUNORI UMEDA, JUN OTA I HISAYUKI KIMURA, *Fusion of multiple ultrasonic sensor data and imagery data for measuring moving obstacle's motion*, Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems
- [10] THOMAS SCHLEGL, THOMAS BRETTERKLIEBER, MARKUS NEUMAYER I HUBERT ZANGL, *Combined Capacitive and Ultrasonic Distance Measurement for Automotive Applications*, IEEE SENSORS JOURNAL, VOL. 11, NO. 11, NOVEMBER 2011
- [11] GERARD GIBBS, HUAMIN JIA I IRFAN MADANI, *Obstacle Detection with Ultrasonic Sensors and Signal Analysis Metrics*, Science Direct – International Conference on Air Transport – INAIR 2017
- [12] ANDRÉ AVELÃS MACHADO DE ARAUJO, RAFAEL LINO DE LIMA, PEDRO CARDOZO DE MELLO, MARIO LUIS CARNEIRO I EDUARDO AOUN TANNURI, *Kalman filter applied to ultrasonic-based level*

sensing, ABCM Symposium Series in Mechatronics - Vol. 5

- [13] ROMAN B. KUC, *Application of Kalman filtering techniques to diagnostic ultrasound*, ULTRASO-NIC IMAGING 1, 105-120 – Department of Electrical Engineering and Computer Science Columbia University New York, NY 10027 (1979)
- [14] PETER S. MAYBECK, *Stochastic models, estimation, and control*, ACADEMIC PRESS, 1979
- [15] JUAN S. GUERRERO GUERRERO, ALDO F. CONTRERAS GONZÁLEZ, JOSE I. HERNÁNDEZ VEGA I LETICIA A. NEIRA TOVAR B, *Instrumentation of an Array of Ultrasonic Sensors and Data Processing for Unmanned Aerial Vehicle (UAV) for Teaching the Application of the Kalman Filter*, 2015 International Conference on Virtual and Augmented Reality in Education
- [16] MATTHEW B. RHUDY, ROGER A. SALGUERO I KEATON HOLAPPA, *A Kalman Filtering tutorial for undergraduate students*, International Journal of Computer Science Engineering Survey (IJCSES) Vol.8, No.1, February 2017
- [17] RICARDO GONZALEZ, *HC-SR04 Ultrasonic sensor – Accuracy and resolution tests*, Llicència Creative Commons
- [18] DATASHEET, *Ultrasonic Ranging Module HC - SR04*
- [19] DUŠAN PONIKVAR, *Programming the STM32F4-Discovery*, University of Ljubljani, Slovenija
- [20] STMICROELECTRONICS, *AN4776 Application note - General-purpose timer cookbook*, STMicroelectronics DocID028459 Rev - May2017
- [21] P. JARVIS, *Lecture 6: Clocks and Timers*, Recurs en línia - users.wpi.edu/
Consultat 15/04/2019

Altres recursos en línia emprats durant el desenvolupament del projecte

Linguee, www.linguee.com/

Recursos d'Octave, octave.org.com/

Stack Overflow, stackoverflow.com/

Overleaf, www.overleaf.com/

Annexos

A Codi de la simulació d'un sensor amb el disseny proposat

```
1  %----- Inicialitzacio parametres -----%
2  x_ = 0;
3  x  = 0;
4  p_ = 0.00000000000001;
5  p  = 0.00000000000001;
6  q  = 0.00001;
7  k  = 0.00000000000001;
8  r  = 0.025;
9  i  = 1;
10 est = 0;
11 cde = [];
12
13
14 %----- Lectura del fitxer -----%
15 mes = dlmread ('dades_x.csv');
16
17 %----- Iteracio -----%
18 x=mes(1);
19
20 for i=1:size(mes)(1)
21     x_ = x;
22     p_ = p + q;
23     k  = p_*(1/(p_+r));
24     est = x_ + k*(mes(i) - x_);
25     p  = (1-k)*p_;
26     x  = est;
27     cde(i) = est;
28     i  = i+1;
29 endfor
30
31 %----- Representacio grafica -----%
32 plot(mes,'g'); hold on;
33 plot(cde,'b');hold off;
34 legend ({"x", "x"}, "location", "east");
35 title ({"x","Abscises: index de dada, Ordenades: dada en centimetres"});
```


B Codi de la simulació d'un sensor amb el processament mixt

```

1  %———— Inicialitzacio de parametres ————%
2  x_ = 0;
3  x  = 0;
4  p_ = 0.00000000000001;
5  p  = 0.00000000000001;
6  q  = 0.00001;
7  q1 = 1;
8  k  = 0.00000000000001;
9  r  = 0.025;
10 i  = 1;
11 est = 0;
12 cde = [];
13
14
15 %———— Lectura del fitxer ————%
16 mes = dlmread ('dades_x.csv');
17
18 %———— Iteracio ————%
19 x=mes(1);
20
21 for i=1:size(mes)(1)
22     if (abs(mes(i)-est)>3)
23         x_ = x;
24         p_ = p + q1;
25         k  = p_*(1/(p_+r));
26         est = x_ + k*(mes(i) - x_);
27         p  = (1-k)*p_;
28         x  = est;
29         cde(i) = est;
30         i  = i+1;
31     else
32         x_ = x;
33         p_ = p + q;
34         k  = p_*(1/(p_+r));
35         est = x_ + k*(mes(i) - x_);
36         p  = (1-k)*p_;
37         x  = est;
38         cde(i) = est;
39         i  = i+1;
40     endif
41 endfor
42
43 %———— Representacio grafica ————%
44 plot(mes,'g'); hold on;
45 plot(cde,'b');hold off;
46 legend ({"x", "x"}, "location", "southwest");
47 title ({"x","Abscises: index de dada, Ordenades: dada en centimetres"});

```


C Codi de la simulació de tres sensors amb el processament mixt

```

1  %———— Inicialitzaci parametres ————%
2  x_1 = 0;
3  x1  = 0;
4  x_2 = 0;
5  x2  = 0;
6  x_3 = 0;
7  x3  = 0;
8
9  p_1 = 0.00000000000001;
10 p1  = 0.00000000000001;
11 p_2 = 0.00000000000001;
12 p2  = 0.00000000000001;
13 p_3 = 0.00000000000001;
14 p3  = 0.00000000000001;
15
16 k1=0.00000000000001;
17 k2=0.00000000000001;
18 k3=0.00000000000001;
19
20 est1 = 0;
21 est2 = 0;
22 est3 = 0;
23
24 q  = 0.00001;
25 q1 = 1;
26 r  = 0.025;
27 i  = 1;
28
29 m=0;
30
31 f1 = 0;
32 f2 = 0;
33 f3 = 0;
34
35 cde = [];
36
37
38 %———— Lectura del fitxer ————%
39 mes1 = dlmread ('sensor_1.csv');
40 mes2 = dlmread ('sensor_2.csv');
41 mes3 = dlmread ('sensor_3.csv');
42
43 %———— Iteraci ————%
44 x1=mes1(1);
45 x2=mes2(1);
46 x3=mes3(1);
47
48 for i=1:1664
49     for f1=0:0
50         if (abs(mes1(i)-est1)>3)
51             x_1 = x1;
52             p_1 = p1 + q1;
53             k1  = p_1*(1/(p_1+r));
54             est1 = x_1 + k1*(mes1(i) - x_1);
55             p1  = (1-k1)*p_1;
56             x1  = est1;

```

```

57     else
58         x_1 = x1;
59         p_1 = p1 + q;
60         k1 = p_1*(1/(p_1+r));
61         est1 = x_1 + k1*(mes1(i) - x_1);
62         p1 = (1-k1)*p_1;
63         x1 = est1;
64     endif
65     f1=1;
66 endfor
67 for f2=0:0
68     if (abs(mes2(i)-est2)>3)
69         x_2 = x2;
70         p_2 = p2 + q1;
71         k2 = p_2*(1/(p_2+r));
72         est2 = x_2 + k2*(mes2(i) - x_2);
73         p2 = (1-k2)*p_2;
74         x2 = est2;
75     else
76         x_2 = x2;
77         p_2 = p2 + q;
78         k2 = p_2*(1/(p_2+r));
79         est2 = x_2 + k2*(mes2(i) - x_2);
80         p2 = (1-k2)*p_2;
81         x2 = est2;
82     endif
83     f2=1;
84 endfor
85 for f3=0:0
86     if (abs(mes3(i)-est3)>3)
87         x_3 = x3;
88         p_3 = p3 + q1;
89         k3 = p_3*(1/(p_3+r));
90         est3 = x_3 + k3*(mes3(i) - x_3);
91         p3 = (1-k3)*p_3;
92         x3 = est3;
93     else
94         x_3 = x3;
95         p_3 = p3 + q;
96         k3 = p_3*(1/(p_3+r));
97         est3 = x_3 + k3*(mes3(i) - x_3);
98         p3 = (1-k3)*p_3;
99         x3 = est3;
100    endif
101    f3=1;
102 endfor
103
104 m=min(est1,est2);
105 cde(i) = min(m,est3);
106 i = i+1;
107 f1=0;
108 f2=0;
109 f3=0;
110 endfor
111
112 %———— Representaci gr fica ————%
113 plot(mes1,'y'); hold on;
114 plot(mes2,'g'); hold on;
115 plot(mes3,'c'); hold on;
116 plot(cde,'k');hold off;

```

```
117 legend ({ "Dades mesurades pel sensor 1", "Dades mesurades pel sensor 2",  
118 "Dades mesurades pel sensor 3", "Q=[1e-5]", "location", "northeast");  
119 title ({ " ", "Circuit i obstacles", " ", "Abscises: index de dada,  
120 Ordenades: dada en centimetres" });
```


D Codi de la solució proposada

Inicialització de paràmetres dins el fitxer main.c :

```

1 //DEFINICI DE PARAMETRES DEL SENSOR D'ULTRASONS
2 /*Estats de prediccio*/
3 __IO float x_1; //estat futur del sensor 1
4 __IO float x1; //estat present del sensor 1
5 __IO float x_2; //estat futur del sensor 2
6 __IO float x2; //estat present del sensor 2
7 __IO float x_3; //estat futur del sensor 3
8 __IO float x3; //estat present del sensor 3
9 /*Matrius de prediccio/actualitzables*/
10 //Covarianca amb l'error inicial
11 __IO float p_1 = 0.000000000001;
12 __IO float p1 = 0.000000000001;
13 __IO float p_2 = 0.000000000001;
14 __IO float p2 = 0.000000000001;
15 __IO float p_3 = 0.000000000001;
16 __IO float p3 = 0.000000000001;
17 //Guany del filtre de Kalman
18 __IO float k1 = 0.000000000001;
19 __IO float k2 = 0.000000000001;
20 __IO float k3 = 0.000000000001;
21 /*Variables constants del filtre*/
22 __IO float q = 0.0001; // Q=[1e-4]
23 __IO float q1 = 1; //Q=[1]
24 __IO float r = 0.025; //Covarianca amb l'error de la mesura
25 __IO float i = 1; //Matriu identitat
26 /*Distancia final dels tres sensors en cm*/
27 __IO float d1;
28 __IO float d2;
29 __IO float d3;
30 /*Variables del bucle*/
31 __IO float m = 0;
32 __IO float f1 = 0;
33 __IO float f2 = 0;
34 __IO float f3 = 0;

```

Presca dels paràmetres externs dins el fitxer stm32fxx_t.c :

```

1
2 //PRESA DE PARAMETRES EXTERNES DEL SENSOR D'ULTRASONS
3 /*Estats de prediccio*/
4 extern __IO float x_1; //estat futur del sensor 1
5 extern __IO float x1; //estat present del sensor 1
6 extern __IO float x_2; //estat futur del sensor 2
7 extern __IO float x2; //estat present del sensor 2
8 extern __IO float x_3; //estat futur del sensor 3
9 extern __IO float x3; //estat present del sensor 3
10 /*Matrius de prediccio/actualitzables*/
11 //Covarianca amb l'error inicial
12 extern __IO float p_1 = 0.000000000001;
13 extern __IO float p1 = 0.000000000001;
14 extern __IO float p_2 = 0.000000000001;
15 extern __IO float p2 = 0.000000000001;

```

```

16 extern __IO float p_3 = 0.000000000001;
17 extern __IO float p3 = 0.000000000001;
18 //Guany del filtre de Kalman
19 extern __IO float k1 = 0.000000000001;
20 extern __IO float k2 = 0.000000000001;
21 extern __IO float k3 = 0.000000000001;
22 /*Variables constants del filtre*/
23 extern __IO float q = 0.0001; // Q=[1e-4]
24 extern __IO float q1 = 1; //Q=[1]
25 extern __IO float r = 0.025; //Covarian a amb l'error de la mesura
26 extern __IO float i = 1; //Matriu identitat
27 /*Distancia final dels tres sensors en cm*/
28 extern __IO float d1;
29 extern __IO float d2;
30 extern __IO float d3;
31 /*Variables del bucle*/
32 extern __IO float m = 0;
33 extern __IO float f1 = 0;
34 extern __IO float f2 = 0;
35 extern __IO float f3 = 0;

```

Bucle de mesura de distància dins el fitxer `stm32fxx_it.c` :

```

1
2 void TIM2_IRQHandler(void)
3 {
4     HAL_GPIO_WritePin(GPIOB,GPIO_PIN_13,GPIO_PIN_SET);
5
6     if( (TIM2->SR & TIM_SR_CC1IF_Msk) != RESET )
7         {USValue = TIM2->CCR1;}
8     if( (TIM2->SR & TIM_SR_CC2IF_Msk) != RESET)
9         {USValue1 = TIM2->CCR2;}
10    if( (TIM2->SR & TIM_SR_CC3IF_Msk) != RESET)
11        {USValue2 = TIM2->CCR3;}
12
13    IRQ_TIM2_counter++;
14
15    if (IRQ_TIM2_counter==3)
16    {
17        //Pel sensor 1
18        for( f1 = 0; f1 = 1 ; f1 = f1 + 1 ){
19            if (abs(USValue*0.0004-x1)>3){
20                x_1=x1;
21                p_1=p1+q1;
22                k1=p_1/(1/(p_1+r));
23                x1=x_1+k1*(USValue*0.0004-x_1);
24                p1=(1-k1)*p_1;
25                d1=x1;
26            }
27            else{
28                x_1=x1;
29                p_1=p1+q;
30                k1=p_1/(1/(p_1+r));
31                x1=x_1+k1*(USValue*0.0004-x_1);
32                p1=(1-k1)*p_1;
33                d1=x1;
34            }
35        }
36    }

```

```

36      //Pel sensor 2
37      for( f2 = 0; f2 = 1 ; f2 = f2 + 1 ){
38          if (abs(USValue1*0.0004-x2)>3){
39              x_2=x2;
40              p_2=p2+q1;
41              k2=p_2+(1/(p_2+r));
42              x2=x_2+k2*(USValue1*0.0004-x_2);
43              p2=(1-k2)*p_2;
44              d2=x2;
45          }
46          else{
47              x_2=x2;
48              p_2=p2+q;
49              k2=p_2+(1/(p_2+r));
50              x2=x_2+k2*(USValue1*0.0004-x_2);
51              p2=(1-k2)*p_2;
52              d2=x2;
53          }
54      }
55      //Pel sensor 3
56      for( f3 = 0; f3 = 1 ; f3 = f3 + 1 ){
57          if (abs(USValue2*0.0004-x3)>3){
58              x_3=x3;
59              p_3=p3+q1;
60              k3=p_3+(1/(p_3+r));
61              x3=x_3+k3*(USValue2*0.0004-x_3);
62              p3=(1-k3)*p_3;
63              d3=x3;
64          }
65          else{
66              x_3=x3;
67              p_3=p3+q;
68              k3=p_3+(1/(p_3+r));
69              x3=x_3+k3*(USValue2*0.0004-x_3);
70              p3=(1-k3)*p_3;
71              d3=x3;
72          }
73      }
74      m = min(d1,d2);
75      USd_k[0]=min(m,d3);
76      if (USd_k[0]>5) USd_k[1]=USd_k[0];
77      if (USd_k[0]>120) USd_k[1]=120;
78      TIM2->EGR = TIM_EGR_UG; // Reinitialize the counter
79  }
80  HAL_GPIO_WritePin(GPIOB,GPIO_PIN_13,GPIO_PIN_RESET);
81  }

```